DEVICE AND METHOD OF PREVENTING PIRATED COPIES OF COMPUTER PROGRAMS

Background Information

A computer program stored on a data carrier can be copied any number of times. Manufacturers of commercial computer programs have therefore attempted to protect their products from unauthorized copying or to ensure that unauthorized copies of their computer programs will not run.

Use of a so-called dongle is a special type of copy protection device. A dongle is a hardware module that must be inserted into a module port of a computer in order to use the respective computer program. The program will not run without the dongle. Although the computer program can be copied as often as desired, it can run only on a computer having a dongle inserted in the module port.

09750423.122800

However, one disadvantage is that the dongle must always remain inserted into the module port of the computer, usually the serial or parallel port, when using the program. This means that one port of the computer is occupied. Furthermore, the program can no longer be used if the dongle is lost. If several computer programs protected in this way are installed on a computer, the user must change dongles if there are too few available ports.

International Patent Publication No. WO 91/1586 describes a dongle that transfers a data file to the computer on installation of the program and modifies the installed program, which cannot run at first, so that it can be used. Then the dongle is no longer necessary and can be removed. This ensures that the computer program will be used on only one computer. However, then the user does not have the option of installing the program on another computer at a later time because the dongle cannot be reused after a single use.

Therefore, an object of the present invention is to provide a device and a method for

EL302702591

preventing bootleg copies of computer programs which will make it possible to use a computer program on different computers at different times.

Summary Of The Invention

The present invention starts with a device having input and output means for a bidirectional data exchange with an electronic computer and a first memory element. The first memory element contains a data file which is to be transferred over the output means to the electronic computer. In the device according to the present invention, there is a second memory element into which data can be written by the input means.

The first and second memory elements are preferably integrated into a memory chip. A nonvolatile semiconductor memory such as a ROM is preferably used as the memory chip.

In an advantageous embodiment of the present invention, the input and output means of the device are designed to correspond to the module port of a computer. The device may have, for example, a jack or a plug that can be inserted into an interface of the computer. In this case, bidirectional data exchange takes place over the assigned interface.

The method according to the present invention includes several process steps. First, the device described above is connected to an electronic computer to permit bidirectional data exchange between the computer and the device. Then a data file containing an electronic key is transferred from the device to the computer. The data file is then only on the computer. As an alternative, the data file may also be copied from the device to the computer and then erased on the device. Following that, a second data file is copied by the computer onto the device. This second data file contains an unambiguously assigned computer identifier. Since the device contains the computer identifier, the electronic key can be transferred back again. The dongle can then be connected to another computer and the key transferred to it. In this way, it is possible to use one computer program on several different computers at different times.

In a preferred embodiment of the method according to the present invention, an encoded enable number is also entered into the computer. On purchasing a program, the customer receives an enable number which has been encoded, i.e., encrypted in the device. On installation of the program, the enable number must be entered. When the program starts, the enable number is decoded with the help of the key and thus certain program modules or different modes are activated. In this way, the seller can provide the customer with an evaluation version or a demo version that can be run only for a certain period of time. If the customer then wants another module or a full version at a later time, he need only be provided with a new enable number. A customer can be provided with this number without any special measures because the enable number can be used only with the proper device.

After the key has been transmitted, the device can be removed again. It is no longer necessary but it is not useless. If the user would like to use the program on another computer at a later time, he must simply connect the device to the computer again. The device then recognizes the computer on the basis of the unambiguous identifier stored in it. After the identifier has been checked, the key can be transferred back again. Then the device can be used with another computer.

Brief Description Of The Drawings

Figure 1 shows a flow chart of how a program with an enable number is installed from a diskette onto the hard drive of a computer.

Figure 2 illustrates how the program is enabled in a computer, i.e., PC, for use with a device according to the present invention, referred to here as a dongle.

Figure 3 illustrates how program modules are enabled with the encoded enable number.

Figure 4 shows how the program can be blocked for use.

Detailed Description

Figure 1 shows a flow chart of how a program with an enable number is installed from a diskette onto the hard drive of a computer.

The installation begins with step 101. Then in step 102 the user is asked whether he would like to install the program. If this is the case, then in step 103 program components are installed from the diskette onto the hard drive of the computer, and then drivers, icons and other modules are installed in step 104. Next in step 105, the user enters the encrypted enable number. If the user decides against installation in step 102 – perhaps the required program modules have already been installed previously – then the program continues directly with step 105. The encrypted enable number that has been entered is stored in a data file in step 106. The installation ends with step 107.

Figure 2 illustrates how the program is enabled in a computer, i.e., PC, for use with a device according to the present invention, referred to here as a dongle.

The program starts with step 201. In step 202 a check is performed to determine whether a dongle is connected. If this is not the case, the program is aborted in step 209. If a dongle is connected, then a check is performed in step 203 to determine whether it contains a valid key. If this is not the case, then a check is performed in step 204 to determine whether the dongle contains the correct PC identifier. The program is aborted if the answer is negative again. If the dongle contains the valid key or the correct PC identifier, then in step 205 the key and the license number of the computer program are copied to the PC. Then in step 206 the key is erased in the dongle. In step 207, the PC identifier is stored in the dongle. The enable process ends with step 208.

Figure 3 illustrates how program modules are enabled with the encoded enable number.

The program is started with step 301. In step 302 a check is performed to determine whether there is a key in the PC. If there is no key, the program is aborted in step 303. If a key is present, the license number of the computer program is read out in step 303. Then the encoded enable number is read out of a data file in step 304. Next in step 305, the enable number is decoded with the license number and with the help of the key. In step 306 the decoded enable number activates the respective program modules. This process ends with step 307.

Figure 4 shows how the program can be blocked for use.

The process begins with step 401. In step 402 a check is performed to determine whether a dongle is connected. If no dongle is connected or otherwise attached, the operation is terminated in step 408. If a dongle is connected, a check is performed in step 403 to determine whether the connected dongle contains the correct PC identifier. If it does not, the operation is aborted. If it does contain the correct PC identifier, the key is copied to the dongle in step 404. Then in step 405, the key in the PC is erased. Following that, the PC identifier in the dongle is erased. The operation ends with step 407.

Since the key is again on the dongle, the dongle can be used to enable the program for running on another computer. This ensures that the program can always run on only one computer at a given time.

A dongle may of course also contain more than one key. For example, if a customer has acquired more than one license, he might either be assigned a corresponding number of dongles each with a separate key or a single dongle with a corresponding number of keys. Multiple dongles with multiple keys could also be issued.

Although the dongle can be connected directly to the computer (PC) on which the program is to be installed, other configurations are also conceivable. In the case of an advantageous embodiment of the present invention, it might be possible to access at least one central dongle containing multiple keys over a network connection, for example. Central software distribution over a large area, in particular throughout a corporation, would thus be possible with the present invention.

If the dongle does not have enough memory for the various PC identifiers and keys, this information could be stored on a data file in an advantageous embodiment. To prevent manipulations here, a check identifier, in particular a checksum, is formed with each access to this data file, and then only this check identifier or checksum is stored in the dongle.

09750823.122800

Software can thus be produced and reproduced in mass production in an advantageous manner without having to apply individual identifiers to the data medium.